Natural Language Processing- Project Report

Topic: Emotion Classification

Report by

Group 27

Farhan Rasheed Chughtai

Bardia Mouhebat

Ananya Kapoor

Abstract

This project focuses on classification of textual data used to express emotion of a person. Our project tries to solve an NLP problem by designing an efficient machine learning model written in python to be able to distinguish these different emotions expressed. We used five machine learning models: Logistic Regression, Naïve Bayes, Random Forest, Feed Forward Neural Network and Bidirectional LSTM recurrent Neural Network to achieve this goal. We identified it as the best performing model based on F-1 score for all the classes.

1.0 Introduction

Emotion detection is an important element of understanding human behavior. Since human emotions affect their decision-making, their interaction with other human beings, and contribute to human intelligence. Recent studies have shown that real-time emotional analysis has been very useful in predicting emotional and behavioral consumer features. Consumer preferences were found to change based on their emotional state, especially in pre-sales and after sales purchases. What was more interesting for us was to use this classification problem to help in understanding customer reviews on a particular product and adjusting this variable into the rating system, instead of manually averaging the rating given by customers. This led us to dig deeper into this topic and we interestingly found that this is not only an NLP problem but also a widely studied topic in psychology, neuroscience, and behavioral science.

Also, we found that this analysis of emotions is highly useful for financial prediction, political decisions, presidential election prediction, personalized recommendation, healthcare (e.g., depression screening), and online teaching (ex, curriculum arrangement) and feedback evaluations.

2.0 Data description

There are two broad categories of human emotions which are positive emotions (happiness/joy Surprise, love) and negative emotions (anger, fear, disgust). The dataset is made up of 20,000 sentences and the distribution of the labels is as shown below:

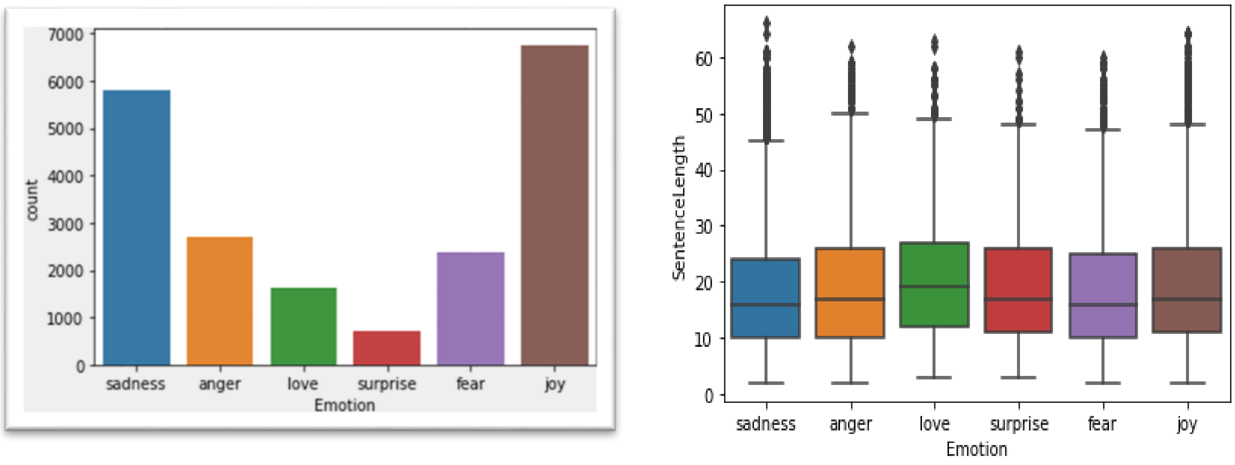

Figure 1: Data distribution of emotions dataset.

3.0 Methodology

The model's we selected for our task of emotion detection are:

1. Logistic Regression
2. Naïve Bayes
3. Random Forest
4. Feed Forward Neural Network
5. Bidirectional LSTM recurrent Neural Network

3.1 Selecting the Models

Out of these we will be using sklearn libraries for model logistic regression, naïve bayes and random forest and we will be using additional NLP methods to implement Neural Network models. The Neural Network model architecture is designed by us, and we use Keras library for model implementation.

Among these five models Random Forest was a new model which we have not covered in class but was a popular classification model, so we tried to use it for this dataset. Also, the Bidirectional LSTM Recurrent Neural Network was chosen because it is especially designed for NLP tasks and this model offers an algorithm which uses long short-term memory (LSTM) recurrent neural network which identifies the relationship between words from start to the end, so it helps to look at the whole sentence together instead of looking at words individually or in a window.

Another benefit of the bidirectional version is that the two-pass system - one from start to the end and then one from the end to the start was immensely beneficial to figure out relationship between words and understanding their context.

3.2 Training the Models

After step 1 of selecting the models, we started by splitting the dataset into train, validation, and test sets under 80:10:10 respectively. This was done before the Data Preprocessing stage of the project so that we do not impact the test set (for example- repetitions of full sentences can occur in test sentence which can give inaccurate results)

3.3 Data Preprocessing

The third step was to fix the disproportion of the label's distributions in our data set. Mainly love and surprise emotions had fewer samples than the rest of the emotions. To fix that we tried the over sampling technique which gave us some good results.

Note: here the over sampling was done only on the training set.

Model	Emotion	F1-Score without Over Sampling	F1-Score with Over Sampling
Random Forest	Surprise	0.75	0.77
Random Forest	Love	0.75	0.78

Table 1: Random Forest results with and without over sampling

After the data preprocessing, we can conclude we were getting better performance for the emotions that had less count of samples, hence we decided to keep oversampling for all the models except the Neural Network.

3.4. Feature Engineering

At this step we started the feature engineering process for our models. Our first three models were trained differently than Neural Network. We tried different inputs for our models.

The first thing we tried was to compare binarized bag of words with normal bag of words. This concept helped us as we observed a small improvement in the performance of the validation set. Therefore, we decided to go with the binarized set of bags of words as our models input.

The next concept we wanted to incorporate was emotion lexicon as features for the first three models. We expected this NLP concept to improve our results, but we found the contrary. We incorporated the following lexicons negative, positive, anger, fear, joy, sadness and surprise, and following Table 2 shows the results.

Model	Accuracy without lexicons	Accuracy with lexicons
Random Forest	0.87	0.80
Logistic Regression	0.90	0.89

Table 2: Results with and without lexicons

From the results above we can see that our overall accuracy of the models decreased and in some models the drop was much larger. One further modification we did was that while creating the bag of words we ignored the stop words, so they were not included in the vocabulary of our bag of words.

The possible reason we later realized could be that many words are common in these lexicons due to the nature of emotions, for example- words like 'joy', 'cheerful' and 'amazingly' were found in both 'joy' and 'surprise' lexicon which was the same word but was used to describe two different emotions – joy and surprise. Therefore, the lexicon features were not helpful as the overlap of words did not add any new information about the training set, rather overlapping confused the models which led to reduced performance in some of our models.

Therefore, what we needed for our model was to try to capture the context of the word as implemented in Neural Network by using word embeddings. Therefore, we decided to limit our work to the bag of words as an input for the first three models and keep the word embeddings as an input for Neural Network.

3.5 Architecture of Neural Network Models:

The neural network models, we did not use any over sampling because it was not making any difference to our results. For our first Neural Network Model which was the feed forward neural network model we used the below architecture:

1. Embedding Layer (Glove 100d Embedding Matrix)
2. Output layer (6, activation=" SoftMax")

It is a simple two-layer architecture with embedding layer populated by the Glove-100d embeddings. We saw that its performance was adequate, but it was not able to beat the logistic regression model so that is why we decided to move on to Bidirectional LSTM Recurrent Neural Network models which are much more powerful and geared towards NLP tasks. For our Bidirectional LSTM Model, we used the below architecture:

1. Embedding Layer (Glove 100d Embedding Matrix)
2. Bidirectional (LSTM (64, return sequences = True)
3. Bidirectional (LSTM (64, return sequences = True)
4. Bidirectional (LSTM (32,)
5. Dense (32, activation = 'relu')
6. Dense (6, activation = 'softmax')

We had 4 hidden layers, three of them being bidirectional LSTM layers and one 'relu' activation layer with 32 hidden nodes.

4.0 Results and Conclusion

Overall, we experimented with five different models for emotion analysis. We try to tune the hyperparameters and our goal was to have the best version of each model based on the validation results. The results for the test set for these models can be seen in table 3. Note: The inputs for the first three models are bag of words and for Neural Networks it is word embeddings.

Model	F1-score	Accuracy	Precision	Recall
Random Forests	0.85	0.87	0.83	0.87
Naïve Bayes	0.77	0.81	0.75	0.79
Logistic Regression	0.87	0.90	0.85	0.89
Feed Forward Neural Network	0.77	0.83	0.77	0.77
Bidirectional LSTM Recurrent Neural Network	0.89	0.93	0.89	0.90

Table 3: Results on the test set for all the models

The reported F1-score, precision and recall are macro-average scores, which are calculated as arithmetic mean of individual classes scores. As shown in table 4, Bidirectional LSTM Recurrent Neural Network has the best performance for the task of emotion analysis.

Interestingly, much simpler models like Logistic Regression and Random Forests also have fairly good performances for this task with 0.87 and 0.85 F1-scores respectively.

Regarding Feed Forward Neural Network, we expected better results than the logistic regression. However, the input for the models is different so it would not be a fair comparison.

The results for Bidirectional LSTM Recurrent Neural Network can be seen in table 4. The model performs best in classifying sentences with 'joy' and 'sadness'. We also observed that the worst performance on sentences with 'surprise' - could be correlated to its low occurrences in our dataset.

Emotion	Precision	Recall	F1-Score	Support
Anger	0.91	0.95	0.93	261
Fear	0.90	0.79	0.84	231
Joy	0.96	0.95	0.96	675
Love	0.85	0.86	0.86	161
Sadness	0.96	0.97	0.97	584
Surprise	0.77	0.86	0.81	88
Macro Average	0.89	0.90	0.89	2000
Weighted Average	0.93	0.93	0.93	2000

Table 4: Results on the test set for Bidirectional LSTM Neural Network

5.0 Future Experiment/Work

We could have tried the BERT Bidirectional Encoder Representations from Transformers language model to see its performance compared with the rest of the models we used in the project. There are several papers using BERT for emotion classification [7]. In addition, we could have tried word embeddings as inputs for the other models.

Due to time constraint and limited resources, we could not experiment with truly deep learning models by that we mean models with at least 15 or 20 layers to see if that would have increased our neural network performance. Another approach we could have tried would see how wide models would perform instead of deep multi-layer models.

We did not test our model on any other data set so for our future work we would test our models on different emotion datasets to see how it would perform on other types of data. One potential dataset could be tweets so we can predict what type of emotion the tweeter is expressing.

6.0 References

1. https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp
2. https://www.kaggle.com/code/aryan7781/emotions-classification-using-lstm
3. https://stackabuse.com/python-for-nlp-movie-sentiment-analysis-using-deep-learning-in-keras
4. https://www.codementor.io/@agarrahul01/multiclass-classification-using-random-forest-on-scikit-learn-library-hkk4lwawu
5. https://www.mygreatlearning.com/blog/bag-of-words/

6. https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
7. BERT-CNN: A Deep Learning Model for Detecting Emotions from Text Ahmed R. Abas, Ibrahim Elhenawy, Mahinda Zidan, and Mahmoud Othman